

ASSIGNMENT 2 : CS432

REPORT

Sachin Yadav 18110148

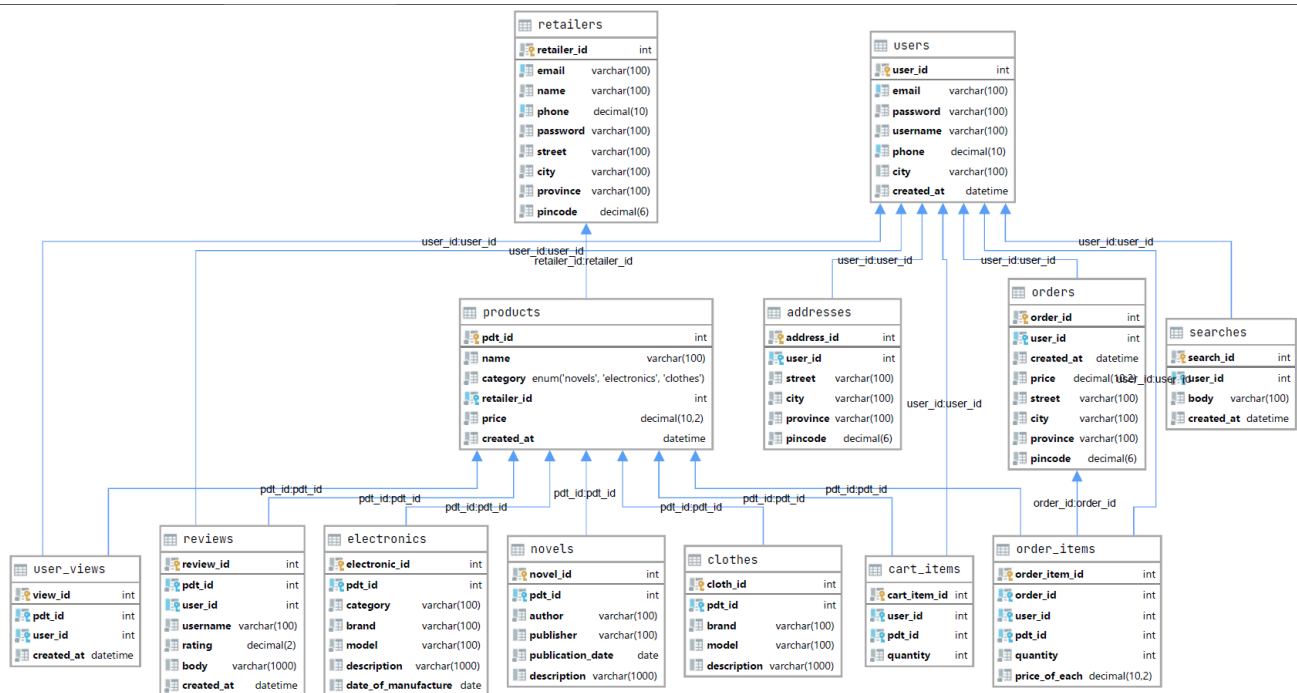
Directory Structure

```
./
├── 01.sql          #the query files for each question numbered from 01 to 21
├── 02.sql
├── 03.sql
├── 04.sql
├── 05.sql
├── 06.sql
├── 07.sql
├── 08.sql
├── 09.sql
├── 10.sql
├── 11.sql
├── 12.sql
├── 13.sql
├── 14.sql
├── 15.sql
├── 16.sql
├── 17.sql
├── 18.sql
├── 19.sql
├── 20.sql
├── 21.sql
├── database_schema.pdf # the Database Schema as generated by JetBrains
DataGrip toop
├── filled-dbdump.sql  #Database Instance with tables filled with dummy
records
├── images             #images of the output of queries and database schema
├── initial-dbdump.sql #Database Instance with empty tables
├── report.md          # the report in MD format, same was converted to PDF
form report.pdf
├── report.pdf         # the PDF of the report
└── schema.sql         #contains the definition of the tables in the database
```

Instructions

- Run `source schema.sql` to initialize a database named 'x' and tables inside it of users, products, etc.
- Run `source 01.sql` to fill the tables with dummy records.

Database Schema



[Generated using JetBrains DataGrip]

- The attributes with blue table sign in front of their name are UNIQUE.
- The Blue Key represents FOREIGN KEY and Gold Key as PRIMARY KEY.

For more detailed Database Schema refer to [schema.sql](#) file

Question 1.

Refer to file [01.sql](#)

Question 2.

Here I deleted a user with email 'niraj@gmail.com'. I have set **ON DELETE CASCADE** for majority of Foreign Keys. In reviews relation, the user_id Foreign Key is allowed to be NULL.

```
UPDATE reviews
SET user_id = NULL, username = 'Anonymous'
WHERE user_id = (SELECT user_id
FROM users
WHERE email = 'niraj@gmail.com');

DELETE FROM users
WHERE email = 'niraj@gmail.com';
```

```
mysql> select * from reviews LIMIT 5;
```

review_id	pdt_id	user_id	username	rating	body	created_at
1	28	4	Niraj Kumar	5	wonderful product, loved it!	2020-12-09 03:36:45
2	19	4	Niraj Kumar	3	NULL	2020-12-11 03:36:45
3	11	4	Niraj Kumar	4	Paper quality was okayish, otherwise wonderful reading	2020-12-21 03:36:45
4	1	2	Anuj Jain	4	NULL	2020-10-09 03:36:45
5	2	3	Kavya Agarwal	3	I found the book to be okayish. The hype was not worth it.	2020-09-11 03:36:45

```
5 rows in set (0.00 sec)

mysql> source 02.sql;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3 Changed: 3 Warnings: 0

Query OK, 1 row affected (0.00 sec)

mysql> select * from reviews LIMIT 5;
```

review_id	pdt_id	user_id	username	rating	body	created_at
1	28	NULL	Anonymous	5	wonderful product, loved it!	2020-12-09 03:36:45
2	19	NULL	Anonymous	3	NULL	2020-12-11 03:36:45
3	11	NULL	Anonymous	4	Paper quality was okayish, otherwise wonderful reading	2020-12-21 03:36:45
4	1	2	Anuj Jain	4	NULL	2020-10-09 03:36:45
5	2	3	Kavya Agarwal	3	I found the book to be okayish. The hype was not worth it.	2020-09-11 03:36:45

```
5 rows in set (0.00 sec)
```

Question 3.

```
UPDATE products
SET price = price * 1.10
where price < 5000 AND pdt_id IN (
    SELECT pdt_id
    FROM user_views
    WHERE created_at >= DATE_SUB(CURDATE(),INTERVAL 3 MONTH)
    GROUP BY pdt_id
    HAVING COUNT(DISTINCT user_id) > 10
);
```

```
mysql> select * from products where price <= 6000;
```

pdt_id	name	category	retailer_id	price	created_at
1	Da Vinci Code	novels	2	450.00	2012-12-22 01:47:46
2	War and Peace	novels	5	1130.00	2011-06-26 01:47:46
3	Ulysses	novels	5	530.00	2019-05-31 01:47:46
4	The Lord of the Rings	novels	2	675.00	2020-12-01 01:47:46
5	Inferno	novels	2	730.00	2013-07-01 01:47:46
6	Lolita	novels	3	430.00	2013-02-15 01:47:46
7	The Lost Symbol	novels	3	478.00	2005-09-24 01:47:46
8	1984	novels	6	550.00	2010-07-22 01:47:46
9	David Copperfield	novels	3	870.00	2006-09-16 01:47:46
10	Angels and Demons	novels	3	699.00	2013-10-01 01:47:46
11	Great Expectations	novels	5	120.00	2007-10-13 01:47:46
14	Neomi Watch R2	electronics	8	3600.00	2008-02-10 01:47:46
24	Leeves Men Jeans	clothes	10	1600.00	2018-10-30 01:47:46

```
13 rows in set (0.01 sec)

mysql> source 03.sql;
Query OK, 4 rows affected (0.07 sec)
Rows matched: 4 Changed: 4 Warnings: 0

mysql> select * from products where price <= 6000;
```

pdt_id	name	category	retailer_id	price	created_at
1	Da Vinci Code	novels	2	450.00	2012-12-22 01:47:46
2	War and Peace	novels	5	1130.00	2011-06-26 01:47:46
3	Ulysses	novels	5	530.00	2019-05-31 01:47:46
4	The Lord of the Rings	novels	2	675.00	2020-12-01 01:47:46
5	Inferno	novels	2	803.00	2013-07-01 01:47:46
6	Lolita	novels	3	473.00	2013-02-15 01:47:46
7	The Lost Symbol	novels	3	478.00	2005-09-24 01:47:46
8	1984	novels	6	550.00	2010-07-22 01:47:46
9	David Copperfield	novels	3	870.00	2006-09-16 01:47:46
10	Angels and Demons	novels	3	699.00	2013-10-01 01:47:46
11	Great Expectations	novels	5	120.00	2007-10-13 01:47:46
14	Neomi Watch R2	electronics	8	3960.00	2008-02-10 01:47:46
24	Leeves Men Jeans	clothes	10	1760.00	2018-10-30 01:47:46

```
13 rows in set (0.01 sec)
```

Here the prices of 5. Inferno, 6. Lolita, 9. David Copperfield, 14. Neomi Watch R2 and 24. Leeves Men Jeans have changed.

Question 4.

```
INSERT INTO addresses (user_id, street, city, province, pincode) VALUES
(1, 'House 52, Mahavir Road, Rohini', 'Delhi', 'Delhi', 110074),
(1, 'Google Office, Nixty Road, Pixis Industrial Complex', 'Bangalore',
'Karnataka', 389252),
(1, 'Room 239, Hostel Chimair, IIT Gandhinagar', 'Gandhinagar', 'Gujarat',
291045);
```

```
mysql> select * from addresses where user_id = 1;
+-----+-----+-----+-----+-----+
| address_id | user_id | street | city | province | pincode |
+-----+-----+-----+-----+-----+
| 1 | 1 | Gangamma Circle, Jalahalli | Bangalore | Karnataka | 560013 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> source 04.sql;
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from addresses where user_id = 1;
+-----+-----+-----+-----+-----+
| address_id | user_id | street | city | province | pincode |
+-----+-----+-----+-----+-----+
| 1 | 1 | Gangamma Circle, Jalahalli | Bangalore | Karnataka | 560013 |
| 21 | 1 | House 52, Mahavir Road, Rohini | Delhi | Delhi | 110074 |
| 22 | 1 | Google Office, Nixty Road, Pixis Industrial Complex | Bangalore | Karnataka | 389252 |
| 23 | 1 | Room 239, Hostel Chimair, IIT Gandhinagar | Gandhinagar | Gujarat | 291045 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Question 5.

```
SELECT DISTINCT phone, email
FROM users
where city = 'Madrid' AND user_id IN (
    SELECT user_id
    FROM orders
    GROUP BY user_id
    HAVING SUM(price) >= 10000
);
```

```
mysql> source 05.sql;
+-----+-----+
| phone | email |
+-----+-----+
| 9972934723 | sachin@gmail.com |
| 9383729512 | anmol@yahoo.com |
+-----+-----+
2 rows in set (0.04 sec)
```

Question 6.

```
SELECT name, pdt_id, user_id
FROM (
    SELECT name, pdt_id
    FROM products
    WHERE name LIKE '%mi%'
)P LEFT OUTER JOIN (
    SELECT DISTINCT user_id, pdt_id
    FROM order_items
)O USING (pdt_id);
```

```
mysql> source 06.sql;
+-----+-----+-----+
| name          | pdt_id | user_id |
+-----+-----+-----+
| Xiaomi S9 Pro | 12     | NULL    |
| Xiom Note 4   | 13     | 5       |
| Neomi Watch R2 | 14     | NULL    |
| Neomi Watch D5 | 16     | 20      |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Question 7.

```
SELECT retailer_id, email
FROM retailers
WHERE city = 'Ahmedabad';
```

```
mysql> source 07.sql;
+-----+-----+
| retailer_id | email                               |
+-----+-----+
| 1           | pvs840@gmail.com                  |
| 2           | cfd@gmail.com                     |
| 6           | ydt@yahoo.com                     |
| 7           | bts@gmail.com                     |
| 10          | pratap@gmail.com                  |
| 11          | puona@rediffmail.com              |
| 15          | tekken@gmail.com                  |
| 16          | fda@gmail.com                     |
| 18          | tiny.ants@rediffmail.com          |
| 19          | tuamir@gmail.com                  |
+-----+-----+
10 rows in set (0.00 sec)
```

Question 8.

```
SELECT *
FROM orders
WHERE user_id = (
    SELECT user_id
    FROM users
    ORDER BY created_at DESC
    LIMIT 1)
ORDER BY created_at DESC
LIMIT 3;
```

```
mysql> source 08.sql;
+-----+-----+-----+-----+-----+-----+-----+
| order_id | user_id | created_at | price | street | city | province | pincode |
+-----+-----+-----+-----+-----+-----+-----+
| 15 | 20 | 2019-11-13 02:44:00 | 1177.00 | Gangamma Circle, Jalahalli | Bangalore | Karnataka | 560013 |
| 14 | 20 | 2018-11-13 02:44:00 | 1205.00 | Gangamma Circle, Jalahalli | Bangalore | Karnataka | 560013 |
| 13 | 20 | 2017-11-13 02:44:00 | 137600.00 | Gangamma Circle, Jalahalli | Bangalore | Karnataka | 560013 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Question 9.

```
SELECT F.user_id, name, pdt_id, category, price
FROM products INNER JOIN (
    SELECT cart_items.user_id, pdt_id
    FROM cart_items INNER JOIN (
        SELECT user_id
        FROM users
        ORDER BY created_at ASC
        LIMIT 2
    ) U USING(user_id)
) F USING (pdt_id);
```

```
mysql> source 09.sql;
+-----+-----+-----+-----+-----+
| user_id | name | pdt_id | category | price |
+-----+-----+-----+-----+-----+
| 1 | The Lord of the Rings | 4 | novels | 675.00 |
| 1 | Leeves Men Jeans | 24 | clothes | 1760.00 |
| 1 | Neomi Watch D5 | 16 | electronics | 11000.00 |
| 1 | Apple iPad | 28 | electronics | 35000.00 |
| 1 | Neomi Watch R2 | 14 | electronics | 3960.00 |
| 1 | Da Vinci Code | 1 | novels | 450.00 |
| 2 | David Copperfield | 9 | novels | 870.00 |
| 2 | Inferno | 5 | novels | 803.00 |
| 2 | Leeves Men Jeans | 24 | clothes | 1760.00 |
+-----+-----+-----+-----+-----+
9 rows in set (0.28 sec)
```

Question 10.

Assumption: By 'after 2010', we mean book published in 2011 (excluding 2010) and coming years.

```
SELECT *
FROM novels
WHERE publication_date >= '2011-01-01';
```

```
mysql> source 10.sql;
```

novel_id	pdt_id	author	publisher	publication_date	description
1	1	Dan Brown	Penguin Books	2016-08-05	NULL
4	4	J.R.R. Tolkien	Pearson	2011-11-21	NULL
7	7	Dan Brown	Penguin Books	2020-09-07	NULL
10	10	Dan Brown	Quarto	2015-01-01	NULL

```
4 rows in set (0.07 sec)
```

Question 11.

```
SELECT pdt_id, e.electronic_id, p.name, price, brand, model, description,
date_of_manufacture
FROM electronics e JOIN products p USING(pdt_id)
WHERE price BETWEEN 10000 AND 20000;
```

```
mysql> source 11.sql;
```

pdt_id	electronic_id	name	price	brand	model	description	date_of_manufacture
12	1	Xiaomi S9 Pro	15000.00	Xiaomi	S9	NULL	NULL
13	2	Xiomi Note 4	20000.00	Xiaomi	Note 4	NULL	NULL
15	4	LG Washing Machine M4	16500.00	LG	M4	NULL	NULL
16	5	Neomi Watch D5	11000.00	Neomi	D5	NULL	NULL
22	11	Realme R1 Pro	16000.00	Realme	R1	NULL	NULL
23	12	Samsung Galaxy S9	20000.00	Samsung	S9	NULL	NULL

```
6 rows in set (0.03 sec)
```

Question 12.

```
SELECT user_id, username, email
FROM
  (SELECT user_id
   FROM order_items INNER JOIN products USING (pdt_id)
   WHERE category = "electronics"
   GROUP BY user_id
   HAVING SUM(quantity) > 3) A
INNER JOIN
  (SELECT user_id
   FROM order_items INNER JOIN products USING (pdt_id)
   WHERE category = "novels"
   GROUP BY user_id
   HAVING SUM(quantity) > 3) B
USING (user_id) JOIN users USING (user_id);
```



```
mysql> source 12.sql;
+-----+-----+-----+
| user_id | username      | email                |
+-----+-----+-----+
|      20 | Captain Levi  | levi84@yahoo.com    |
+-----+-----+-----+
1 row in set (0.11 sec)
```

Question 13.

```
SELECT pdt_id, electronic_id, name, electronics.category,
electronics.brand, electronics.model, price
FROM electronics JOIN products USING (pdt_id)
WHERE electronics.pdt_id = products.pdt_id AND electronics.category =
"Laptop"
ORDER BY price ASC;
```

```
mysql> source 13.sql;
+-----+-----+-----+-----+-----+-----+
| pdt_id | electronic_id | name                | category | brand | model | price |
+-----+-----+-----+-----+-----+-----+
| 19 | 8 | Dell Inspiron D3 | Laptop | Dell | D3 | 45000.00 |
| 20 | 9 | Dell Inspiron G4 | Laptop | Dell | G4 | 55000.00 |
| 17 | 6 | HP Pavilion G6 | Laptop | HP | G6 | 56000.00 |
| 18 | 7 | Acer Predator Helio | Laptop | Acer | Helio93 | 74500.00 |
| 21 | 10 | Apple macbook Air | Laptop | Apple | air | 85000.00 |
| 25 | 13 | Apple macbook Pro | Laptop | Apple | Pro | 90000.00 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Question 14.

```
SELECT *
FROM products
WHERE created_at >= '2011-11-12 00:00:00';
```

```
mysql> source 14.sql;
```

pdt_id	name	category	retailer_id	price	created_at
1	Da Vinci Code	novels	2	450.00	2012-12-22 01:47:46
3	Ulysses	novels	5	530.00	2019-05-31 01:47:46
4	The Lord of the Rings	novels	2	675.00	2020-12-01 01:47:46
5	Inferno	novels	2	803.00	2013-07-01 01:47:46
6	Lolita	novels	3	473.00	2013-02-15 01:47:46
10	Angels and Demons	novels	3	699.00	2013-10-01 01:47:46
12	Xiaomi S9 Pro	electronics	1	15000.00	2015-02-09 01:47:46
15	LG Washing Machine M4	electronics	7	16500.00	2015-12-01 01:47:46
16	Neomi Watch D5	electronics	8	11000.00	2020-09-07 01:47:46
17	HP Pavillion G6	electronics	7	56000.00	2016-01-06 01:47:46
18	Acer Predator Helio	electronics	1	74500.00	2015-05-01 01:47:46
20	Dell Inspiron G4	electronics	9	55000.00	2013-02-26 01:47:46
21	Apple macbook Air	electronics	10	85000.00	2019-12-27 01:47:46
22	Realme R1 Pro	electronics	1	16000.00	2011-11-29 01:47:46
24	Leeves Men Jeans	clothes	10	1760.00	2018-10-30 01:47:46
25	Apple macbook Pro	electronics	9	90000.00	2020-12-30 01:47:46
26	Apple iPad Pro	electronics	9	60000.00	2020-11-30 01:47:46
27	Apple iPad Air	electronics	9	54000.00	2021-01-30 01:47:46
28	Apple iPad	electronics	9	35000.00	2021-02-02 01:47:46

19 rows in set (0.00 sec)

Question 15.

```
SELECT name, novels.*, price
FROM novels JOIN products using (pdt_id)
WHERE author = "Dan Brown";
```

```
mysql> source 15.sql;
```

name	novel_id	pdt_id	author	publisher	publication_date	description	price
Da Vinci Code	1	1	Dan Brown	Penguin Books	2016-08-05	NULL	450.00
Inferno	5	5	Dan Brown	Wiley	2008-12-27	NULL	803.00
The Lost Symbol	7	7	Dan Brown	Penguin Books	2020-09-07	NULL	478.00
Angels and Demons	10	10	Dan Brown	Quarto	2015-01-01	NULL	699.00

4 rows in set (0.04 sec)

Question 16.

```
SELECT DISTINCT user_id, username, email, phone
FROM users
WHERE user_id in (
    SELECT user_id
    FROM cart_items
    WHERE quantity < 5
);
```

```
mysql> source 16.sql;
+-----+-----+-----+-----+
| user_id | username | email | phone |
+-----+-----+-----+-----+
| 1 | Sachin Yadav | sachin@gmail.com | 9972934723 |
| 3 | Kavya Agarwal | kavya@rediffmail.com | 2035753956 |
| 2 | Anuj Jain | anuj@yahoo.com | 7502740385 |
| 6 | Dhiraj Sinha | dhiraj@outlook.com | 3829893782 |
| 7 | Kunal Kumar | kunal@iitgn.ac.in | 6483026593 |
| 9 | Sushant Kumar | sushant.kumar@gmail.com | 3792052942 |
| 10 | Kalyan G. Reddy | kalyan.reddy@iitgn.ac.in | 3920395343 |
| 11 | Sushant Goyal | sushant.goyal@gmail.com | 7583925935 |
| 12 | Andy Murray | andy043@gmail.com | 4832840284 |
| 13 | Nicole Ray | nicole58@yahoo.com | 8302643252 |
| 14 | Jon Duse | jon380@rediffmail.com | 7296937501 |
+-----+-----+-----+-----+
11 rows in set (0.07 sec)
```

Question 17.

```
SELECT *
FROM orders
WHERE order_id = (
    SELECT order_id
    FROM order_items
    GROUP BY order_id
    ORDER BY SUM(quantity) DESC
    LIMIT 1);
```

```
mysql> source 17.sql;
+-----+-----+-----+-----+-----+-----+-----+
| order_id | user_id | created_at | price | street | city | province | pincode |
+-----+-----+-----+-----+-----+-----+-----+
| 9 | 3 | 2018-11-13 02:44:00 | 3250.00 | Shanti Park, Mira Road | Mumbai | Maharashtra | 401107 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Question 18.

```
SELECT *
FROM products
WHERE created_at >= DATE_SUB(CURDATE(), INTERVAL 10 DAY);
```

```
mysql> source 18.sql
+-----+-----+-----+-----+-----+-----+
| pdt_id | name | category | retailer_id | price | created_at |
+-----+-----+-----+-----+-----+-----+
| 27 | Apple iPad Air | electronics | 9 | 54000.00 | 2021-01-30 01:47:46 |
| 28 | Apple iPad | electronics | 9 | 35000.00 | 2021-02-02 01:47:46 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Question 19.

```

SELECT DISTINCT retailer_id
FROM products
WHERE pdt_id in (
    SELECT pdt_id
    FROM order_items
    WHERE user_id = 1
);

```

```

mysql> source 19.sql;
+-----+
| retailer_id |
+-----+
|          2 |
|          9 |
|          3 |
+-----+
3 rows in set (0.00 sec)

```

Question 20.

Assumption:: The prices in the original products tables will remain same. Only the prices in holi_Deals change.

```

CREATE TABLE holi_Deals
SELECT *
FROM products
WHERE created_at >= DATE_SUB(CURDATE(),INTERVAL 100 DAY);

UPDATE holi_Deals
SET price = price * 0.85;

```

```

mysql> source 20.sql;
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

Query OK, 5 rows affected (0.00 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from holi_Deals;
+-----+-----+-----+-----+-----+-----+
| pdt_id | name                | category | retailer_id | price  | created_at |
+-----+-----+-----+-----+-----+-----+
| 4      | The Lord of the Rings | novels   | 2          | 573.75 | 2020-12-01 01:47:46 |
| 25     | Apple macbook Pro    | electronics | 9          | 76500.00 | 2020-12-30 01:47:46 |
| 26     | Apple iPad Pro       | electronics | 9          | 51000.00 | 2020-11-30 01:47:46 |
| 27     | Apple iPad Air       | electronics | 9          | 45900.00 | 2021-01-30 01:47:46 |
| 28     | Apple iPad           | electronics | 9          | 29750.00 | 2021-02-02 01:47:46 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Question 21.

About the recommendation algorithm

- Prepare a list of all the products that matches with the past searches of the user.
[Descending Order of the time of search]
- Remove all the items for the above list, which the user has already bought after searching.
- Here, I thought that a user have searched for a product and bought one. So I need not recommend the same product that he/she bought again, instead I show the other products to which his/her searches matches.

```
SELECT pdt_id, name, price
FROM (
    SELECT user_id, searches.created_at, pdt_id
    FROM searches, products
    WHERE name LIKE CONCAT('%', body, '%')
) T INNER JOIN products USING(pdt_id)
WHERE T.user_id = 1 AND pdt_id NOT IN (
    SELECT pdt_id
    FROM order_items
    WHERE user_id = 1
)
GROUP BY pdt_id
ORDER BY MAX(T.created_at) DESC
LIMIT 10;
```

```
mysql> source 21.sql;
```

pdt_id	name	price
26	Apple iPad Pro	43350.00
27	Apple iPad Air	39015.00
28	Apple iPad	25287.50
21	Apple macbook Air	85000.00
25	Apple macbook Pro	65025.00
14	Neomi Watch R2	3960.00
16	Neomi Watch D5	11000.00
12	Xiaomi S9 Pro	15000.00
20	Dell Inspiron G4	55000.00
1	Da Vinci Code	450.00

10 rows in set (0.00 sec)

Explanation: (Dates given in Descending -- User 1)

- Searched "iPad" on 26 Nov 2020.
- Searched "Apple" on 20 Nov 2020.
- Ordered "Lolita" on 10 Nov 2020.
- Searched "Lolita" on 1 November 2020.
- Searched "watch" on 20 Oct 2020.

- Searched "Xiaomi" on 15 Oct 2020.
- Ordered "Dell Inspiron D3" on 15 Oct 2020.
- Searched "Dell" on 20 Sept 2020.
- Searched "Xiaomi" on 12 Aug 2020.
- Ordered "Inferno" on 07 Jan 2010.
- Searched for "Inferno" on 06 Jan 2010.
- Searched for "Vinci Code" on 06 Jan 2010.